บทที่ 2

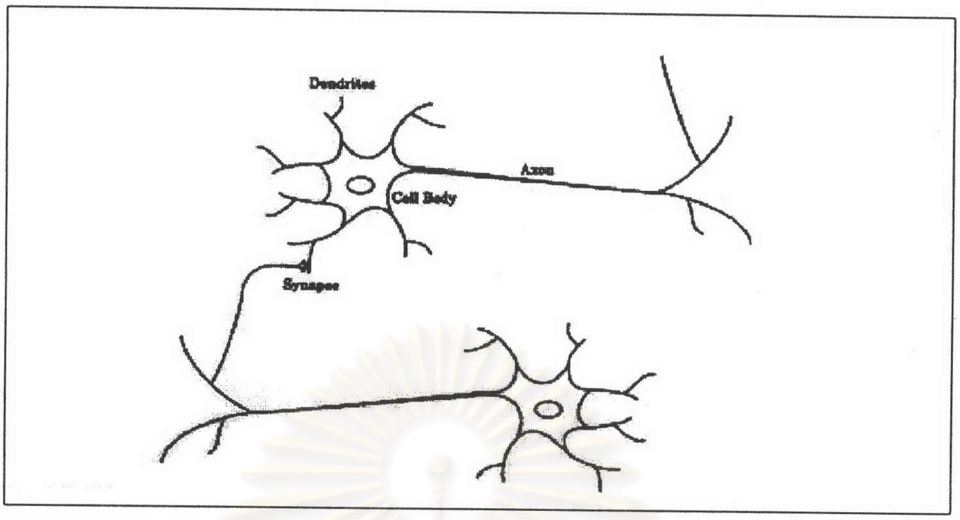
ความรู้เบื้องต้นและทฤษฎีเกี่ยวกับเครือข่ายนิวรอน

เครือข่ายนิวรอนเป็นอัลกอริทึมที่พัฒนาเพื่อเลียนแบบกลวิธีการประมวลผลข้อมูลของเซลประสาทในสมอง เพื่อนำมาใช้ในการจัดการปัญหาที่ไม่ต้องการขบวนการนิยามอย่างเด่นชัดเหมือนกับปัญหาทางคณิตศาสตร์โดยทั่วๆไป ในทางตรงกันข้ามการแก้ปัญหาจะใช้วิธีการเรียนรู้และรวบรวมข้อมูลตัวอย่างของปัญหา รวมทั้งคำตอบสำหรับปัญหานั้นๆ ไว้ในฐานข้อมูลของระบบ และระบบจะพยายามปรับปรุงความรู้ภายในเพื่อให้สามารถหาคำตอบของปัญหาที่มีลักษณะคล้ายกับตัวอย่างที่ได้เคยเรียนรู้

ในบทนี้จะกล่าวถึงความรู้เบื้องต้น และทฤษฎีที่เกี่ยวกับเครือข่ายนิวรอน รวมทั้งการปรับปรุงกฎการเรียนรู้ที่ใช้ในการฝึกเครือข่าย เพื่อให้การลู่เข้าหาคำตอบเป็นไปได้อย่างรวดเร็ว

2.1 <u>ลักษณะกระบวนการรับรู้ของเซลประสาทในสมอง</u>

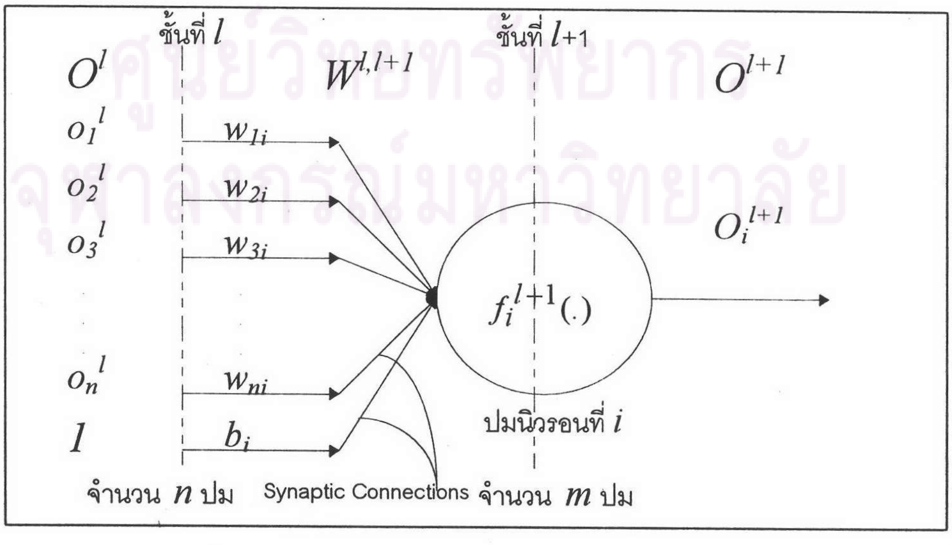
ภายในสมองมนุษย์ประกอบด้วยหน่วยประมวลผลข้อมูลขนาดเล็ก ที่เรียกว่า เซลสมอง หรือ นิวรอน (Neurons) ซึ่งจะมีประมาณ 10^{11} หน่วย ในเซลสมองแต่ละหน่วยดังแสดงในรูปที่ 2.1 ประกอบไปด้วย Dendrites, Cell Body และ Axon โดย Dendrites โดยเป็นใยประสาทที่อยู่รอบๆ ตัวเซล ซึ่งหน้าที่ในการรับสัญญาณเข้าสู่ตัวเซล Axon และส่งผ่านสัญญาณที่ประมวลได้จากตัวเซล (Cell Body) ไปยังเซลสมองหน่วยอื่น ส่วนของการเชื่อมโยงระหว่าง Axon ของเซลสมองหนึ่งไปยัง Dendrites ของเซลสมองอื่นเรียกว่า Synapse โดยเป็นตัวกลางในการส่งข้อมูลที่ได้จากการประมวลผลของเซลสมองแต่ละตัว ในเซลสมองแต่ละเซลจะมี Synapse ที่เชื่อมโยงอยู่โดยประมาณ 10^4 เส้น ซึ่งทำหน้าที่ช่วยในการคิดและคำนวน เพื่อให้ได้ผลของข้อมูลที่ต้องและแม่นยำ



รูปที่ 2.1 แสดงลักษณะของเซลสมองอย่างง่าย

2.2 รูปแบบของสมการทางคณิตศาสตร์ที่ใช้แทนเครือข่ายนิวรอน

จากลักษณะการทำงานของเซลประสาทสมองของมนุษย์ ดังกล่าวในหัวข้อ 2.1 ข้างต้น ในปี ค.ศ. 1943 McCulloch และ Pitts [33] ได้เป็นผู้ริเริ่มในการจำลองลักษณะการทำงานของเซลประสาทสมองโดยใช้สมการทางคณิตศาสตร์อย่างง่าย โดยมีข้อจำกัดคือ สามารถจัดการเฉพาะกับปัญหาที่เป็นลักษณะทางตรรกศาสตร์อย่างง่าย เช่น ปัญหาตรรกศาสตร์ NOR OR และ AND เป็นต้น และจากจุดเริ่มต้นนี้ได้มีการพัฒนาแบบจำลองดังกล่าวจนเป็นที่ยอมรับและนิยมใช้กันอย่างแพร่หลายในเวลาต่อมา ดังแสดงแบบจำลองที่ใช้แทนปมนิวรอนในรูปที่ 2.2



รูปที่ 2.2 แสดงแบบจำลองที่ใช้แทนปมนิวรอน

จากรูปที่ 2.2 ปมนิวรอนประกอบไปด้วยส่วนสำคัญ 3 ส่วนได้แก่

1. หน่วยประมวลผล (Processing Elements) เรียกว่า “ปม” หรือ “Node”

2. Synaptic Connections เป็นส่วนเชื่อมต่อระหว่างปมนิวรอนในแต่ละชั้นสำหรับการส่งข้อมูลที่ประมวลผลได้จากปมนิวรอนหนึ่งๆ ไปยังปมอื่น

3. ค่าน้ำหนัก (Weight) และค่าไบแอส (Bias) ที่กำกับในแต่ละแขน ซึ่งทำหน้าที่ในการขยายหรือลดขนาดของสัญญาณที่เข้าสู่ปมนิวรอน ซึ่งสอดคล้องกับกฎการเรียนรู้ที่ใช้ในการฝึกเครือข่ายนิวรอน

จากรูปที่ 2.2 สามารถเขียนแทนด้วยสมการทางคณิตศาสตร์ได้ดังนี้

$$O^{l+1} = F^{l+1}(W^{l,l+1}O^{l} + B^{l+1}) \quad \text{.............................(2.1)}$$

<u>กำหนด</u>

l ลำดับชั้นภายในเครือข่ายนิวรอน

n จำนวนปมนิวรอนในชั้นที่ l

m จำนวนปมนิวรอนในชั้นที่ $l+1$

$W^{l,l+1}$ เมตริกซ์ค่าน้ำหนักของเครือข่ายนิวรอน ซึ่งเชื่อมระหว่างชั้นที่ l และ ขนาด $n \times m$

$$W^{l,l+1} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ w_{n1} & w_{n2} & \cdots & w_{nm} \end{bmatrix} \quad \text{.......................................(2.2)}$$

โดยที่ w_{nm} แทนค่าน้ำหนักซึ่งเชื่อมต่อระหว่างปมนิวรอนที่ n ในชั้นที่ l และปมนิวรอนที่ m ในชั้นที่ $l+1$

B^{l+1} เวกเตอร์ค่าไบแอสของเครือข่ายนิวรอล ที่กำกับในชั้นที่ $l+1$ ขนาด $m \times 1$

$$B^{l+1} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{bmatrix} \text{...(2.3)}$$

โดยที่ b_m แทนค่าไบแอสซึ่งเชื่อมต่อปมนิวรอนที่ m ในชั้นที่ $l+1$

O^{l+1} เวกเตอร์ค่าสัญญาณเอาต์พุตจากปมนิวรอนในชั้นที่ $l+1$ ขนาด $m \times 1$

$$O^{l+1} = \begin{bmatrix} o_1^{l+1} \\ o_2^{l+1} \\ \dots \\ o_m^{l+1} \end{bmatrix} \text{..(2.4)}$$

โดยที่ o_m^{l+1} แทนค่าสัญญาณเอาต์พุตจากปมนิวรอนที่ m ในชั้นที่ $l+1$

$F^{l+1}(.)$ เวกเตอร์ค่า Activation Function ของปมนิวรอนในชั้นที่ $l+1$ ขนาด $m \times 1$

$$F^{l+1}(.) = \begin{bmatrix} f_1^{l+1}(.) \\ f_2^{l+1}(.) \\ \dots \\ f_m^{l+1}(.) \end{bmatrix} \text{..(2.5)}$$

โดยที่ $f_m^{l+1}(.)$ แทน Acivation Function ของปมนิวรอนที่ m ในชั้นที่ $l+1$

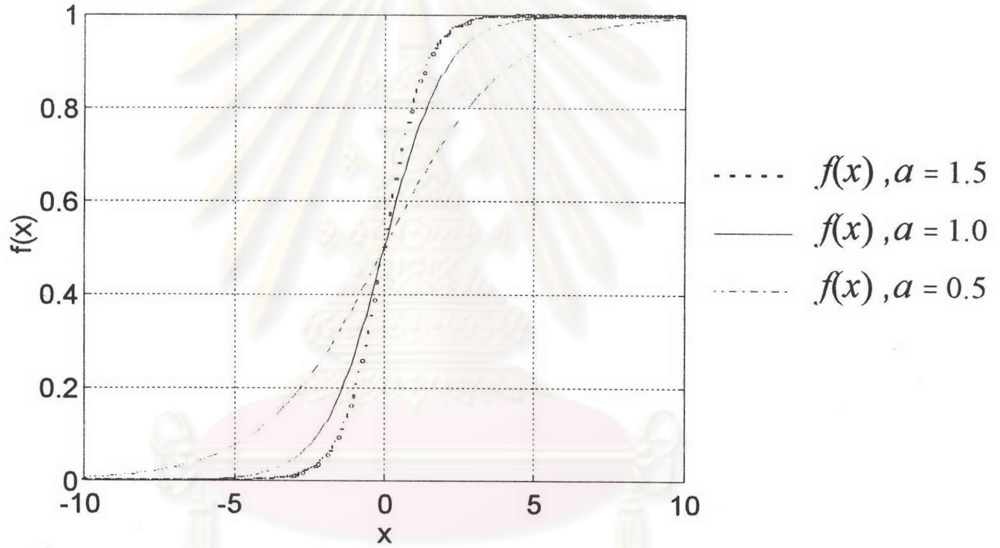
$f(x)$ ฟังก์ชันซิกมอยด์ (Sigmoid Acivation Function)

$$f(x) = \frac{1}{1+e^{-a.x}} \text{..(2.6)}$$

โดยที่ x สัญญาณอินพุตที่เข้าสู่ฟังก์ชันซิกมอยด์

a ค่าสัมประสิทธิ์ของฟังก์ชันซิกมอยด์

ฟังก์ชันซิกมอยด์เป็นฟังก์ชันต่อเนื่องสามารถหาค่าอนุพันธ์ได้ตลอดช่วงและมีค่าไม่ลดลง โดยมีค่าอยู่ระหว่าง 0 และ 1 เมื่อค่าสัญญาณเข้าสู่ Activation Function : x มีค่าเปลี่ยนจาก $-\infty$ ไปยังค่า ∞ ตามลำดับ ค่าสัมประสิทธิ์ของฟังก์ชันซิกมอยด์เป็นค่าที่กำหนดลักษณะของความชันของกราฟ โดยถ้าค่าสัมประสิทธิ์มีค่าน้อยช่วงการเปลี่ยนแปลงค่าจาก 0 ไป 1 จะกว้าง ในทางกลับกันถ้ามีค่ามากระยะการเปลี่ยนแปลงดังกล่าวจะแคบ ข้อดีของช่วงการเปลี่ยนที่กว้างคือ เป็นการเพิ่มช่วงการทำงานของปมนิวรอน และป้องกันสัญญาณออกถึงจุดอิ่มตัวเร็วเกินไปซึ่งจะทำให้เครือข่ายนิวรอนไม่สามารถเรียนรู้ต่อไปได้ จากลักษณะของฟังก์ชันดังกล่าวทำให้สามารถใช้เทคนิคการออพติไมซ์แบบเกรเดียนท์สร้างอัลกอลิทึมการเรียนรู้ของเครือข่ายนิวรอนได้กราฟของฟังก์ชันซิกมอยด์ซึ่งมีค่าสัมประสิทธิ์ a ต่างๆกัน แสดงดังในรูปที่ 2.3



***<u>รูปที่ 2.3</u>* แสดงกราฟของฟังก์ชันซิกมอยด์** *(Sigmoid Function)*

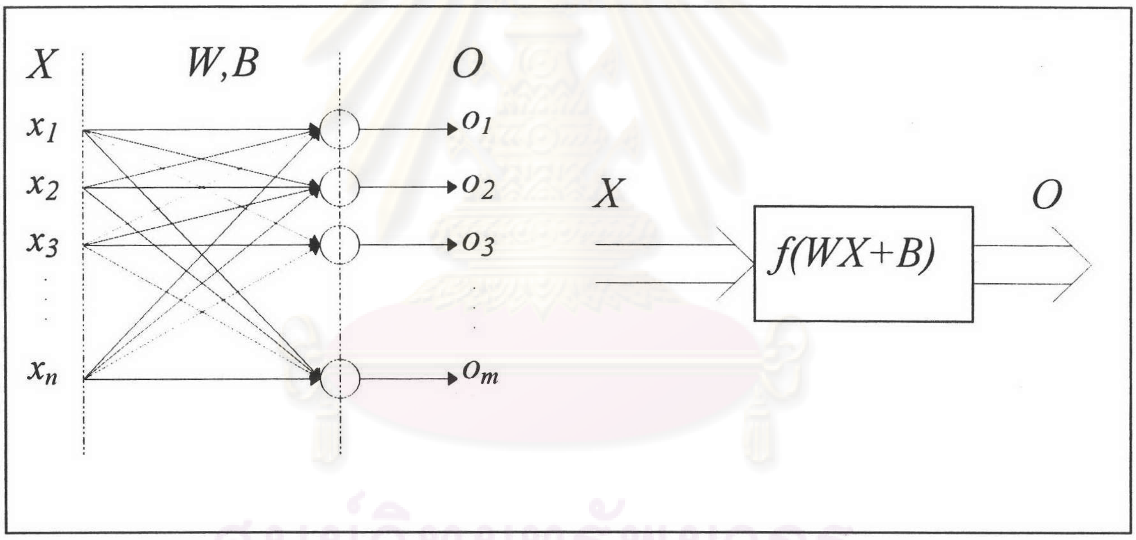
เครือข่ายนิวรอนประกอบด้วยปมนิวรอนซึ่งเรียงตัวอยู่เป็นชั้นๆ (Layer) โดยรับข้อมูลที่ต้องการประมวลผลจากชั้นอินพุต (Input Layer) ข้อมูลดังกล่าวจะถูกส่งต่อไปยังปมนิวรอนแต่ละปมในชั้นอินพุต ปมนิวรอนดังกล่าวจะประมวลผลหาค่าสัญญาณออกและส่งผ่าน Synaptic Connections ไปยังปมนิวรอนในชั้นถัดมา โดยข้อมูลความรู้ต่างๆที่ประมวลได้จะถูกเก็บอยู่ในรูปของค่าน้ำหนักและค่าไบแอสซึ่งกำกับในแต่ละแขนของ Synaptic Connections ลักษณะการทำงานดังกล่าวจะกระทำซ้ำๆกัน ในแต่ละชั้นจนถึงชั้นเอาต์พุต (Output Layer) แล้วทำการส่งข้อมูลที่ประมวลได้ทั้งหมดออกมาเป็นคำตอบของปัญหาที่ต้องการต่อไป

2.3 ชนิดของการเชื่อมต่อภายในเครือข่ายนิวรอน

โดยทั่วไปลักษณะการเชื่อมต่อภายในเครือข่ายนิวรอน สามารถแบ่งออกได้เป็น 2 ชนิดคือ

2.3.1 เครือข่ายนิวรอนแบบป้อนไปข้างหน้า (Feedforward Neural Network)

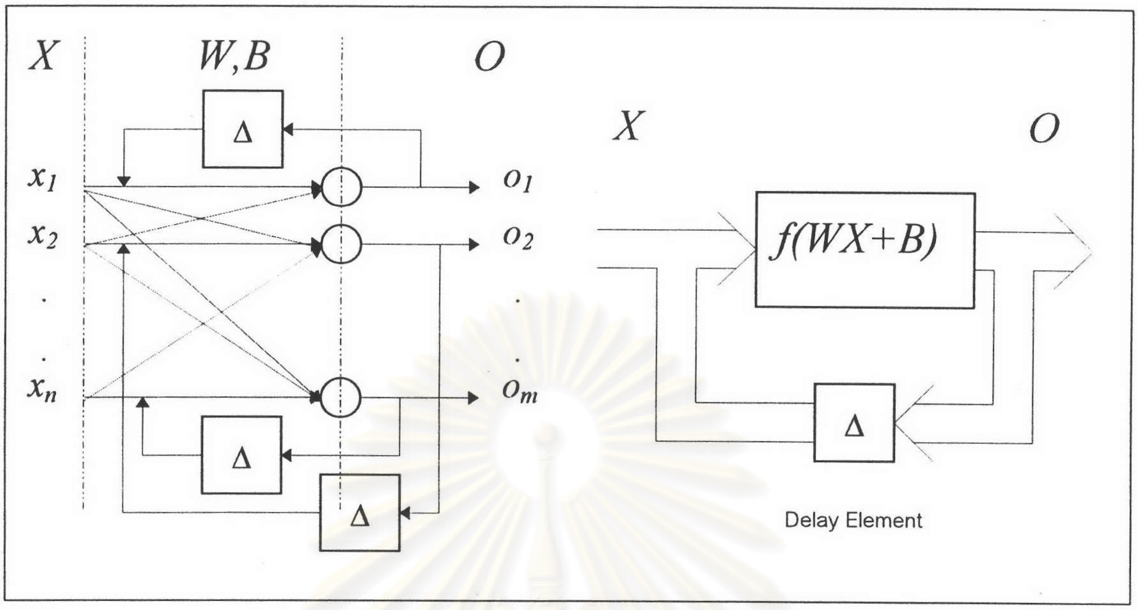
เป็นเครือข่ายที่การประมวลผลจะอาศัยชุดของข้อมูลปัจจุบัน และส่งค่าที่ประมวลผลได้ไปยังชั้นถัดๆไป กล่าวคือ ข้อมูลที่ใช้ในการฝึกเครือข่ายเมื่อถูกป้อนเข้ามา จะส่งถูกส่งไปยังปมนิวรอนชั้นอินพุต ชั้นซ่อนภายใน และออกสู่ชั้นเอาต์พุต ข้อมูลที่ได้จึงเป็นผลจากการประมวลผลข้อมูลโดยรวมทั้งหมด สามารถแสดงแบบจำลองเครือข่ายแบบป้อนไปข้างหน้า ได้ดังรูปที่ 2.4



รูปที่ 2.4 *แสดงลักษณะของเครือข่ายนิวรอนแบบป้อนไปข้างหน้า*

2.3.2 เครือข่ายนิวรอนแบบมีการป้อนกลับ (Feedback Neural Network หรือ Recurrent Neural Network)

เป็นเครือข่ายนิวรอนที่จะอาศัยทั้งข้อมูลในปัจุบันและข้อมูลที่มีการประวิงเวลาในการนำมาใช้ในการบวนการประมวลผลของเครือข่ายนิวรอน สามารถแสดงแบบจำลองเครือข่ายนิวรอนแบบมีการป้อนกลับ ได้ดังรูปที่ 2.5



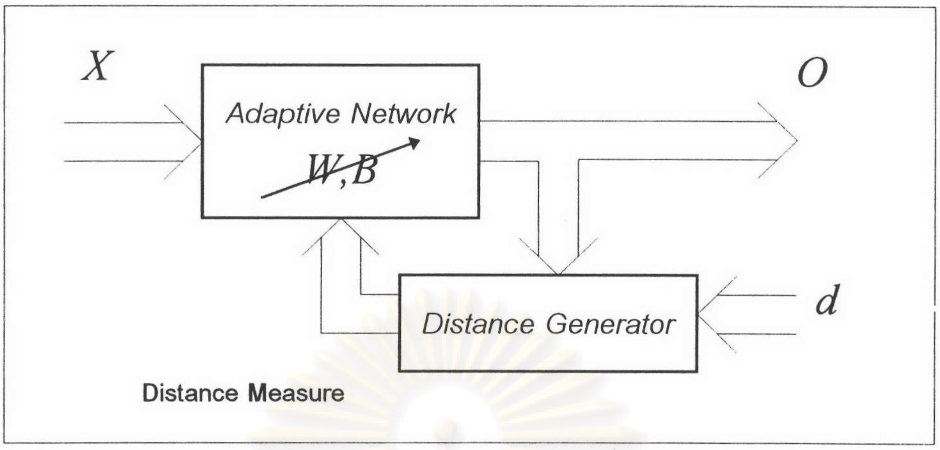
<u>รูปที่ 2.5</u> แสดงลักษณะเครือข่ายนิวรอนแบบมีการป้อนกลับ

2.4 <u>กลวิธีการเรียนรู้และการปรับค่า</u>

ในการฝึกเครือข่ายนิวรอนให้เรียนรู้เพื่อหาคำตอบของปัญหาหนึ่งๆนั้น อาศัยชุดของปัญหาและคำตอบของปัญหาที่ถูกต้อง เพื่อนำมาใช้ในการฝึกเครือข่ายนิวรอนให้สามารถสร้างคำตอบของปัญหาให้คล้ายคลึงกับชุดของข้อมูลที่ได้รับการฝึกให้เรียนรู้ โดยทั่วไปแล้วสามารถจำแนกกลวิธีการเรียนรู้ของเครือข่ายนิวรอนออกได้เป็น 2 ประเภท คือ

2.4.1 <u>การเรียนรู้โดยอาศัยการแนะนำ (Supervised Learning)</u>

เป็นการฝึกเครือข่ายนิวรอนให้เลียนแบบชุดข้อมูลตามที่กำหนด และสามารถสร้างสัญญาณออกได้คล้ายคลึงกับชุดข้อมูลที่ป้อนให้เรียนรู้นั้นๆ ดังแสดงในรูปที่ 2.6

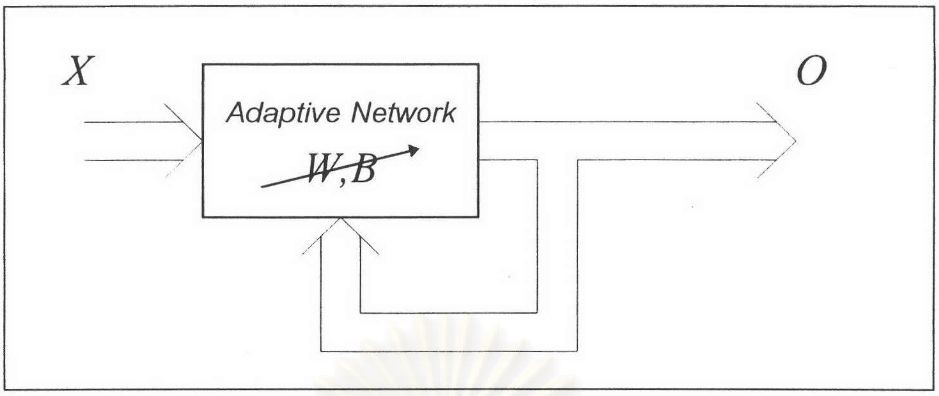


รูปที่ 2.6 แสดงลักษณะการเรียนรู้โดยอาศัยการแนะนำ

ชุดของข้อมูลที่ใช้ในการฝึกเครือข่ายนิวรอน (Training Set) จะประกอบไปด้วยชุดข้อมูลสัญญาณอินพุต (*X*) และข้อมูลสัญญาณเป้าหมาย (*d*) จากรูปที่ 2.6 เมื่อป้อนชุดข้อมูลสัญญาณอินพุตให้กับเครือข่ายนิวรอน เครือข่ายนิวรอนจะประมวลผลสัญญาณดังกล่าวเพื่อให้สัญญาณออกจากเครือข่ายสอดคล้องกับสัญญาณเป้าหมายตามที่กำหนดในชุดข้อมูลที่ใช้ในการฝึก (Training Set) โดยอาศัยค่าความแตกต่างระหว่างสัญญาณเอาต์พุตจากเครือข่าย (*O*) และสัญญาณเป้าหมายในการปรับค่าพารามิเตอร์ของเครือข่ายนิวรอน ได้แก่ ค่าน้ำหนัก (Weight) และค่าไบแอส (Bias) เพื่อให้เครือข่ายนิวรอนสามารถเรียนรู้และจดจำข้อมูลสัญญาณอินพุต และสร้างสัญญาณเอาต์พุตได้สอดคล้องกับข้อมูลสัญญาณเป้าหมายตามที่กำหนดในชุดของข้อมูลที่ใช้ฝึกดังกล่าว

2.4.2 การเรียนรู้โดยไม่อาศัยการแนะนำ (Unsupervised Learning)

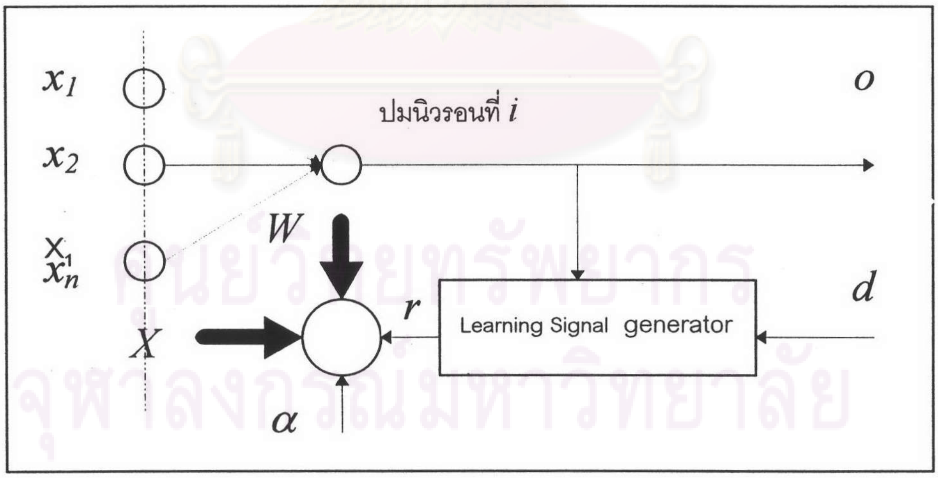
เป็นการเรียนรู้ที่อาศัยชุดข้อมูลสัญญาณอินพุต (*X*) โดยไม่มีชุดข้อมูลสัญญาณเป้าหมายเพื่อใช้ฝึกเครือข่าย การปรับค่าพารามิเตอร์ของเครือข่ายนิวรอนอาศัยสัญญาณเอาต์พุตจากเครือข่าย (*O*) อย่างเดียว ดังแสดงในรูปที่ 2.7 วิธีการปรับมีหลายวิธี ซึ่งในแต่ละวิธีมีวัตถุประสงค์เพื่อให้เครือข่ายนิวรอนสามารถจดจำและจำแนกรูปแบบของข้อมูลที่เข้ามาได้ นิยมประยุกต์ใช้กับปัญหาในการจำแนกชุดข้อมูล (Clustering) ฯลฯ



รูปที่ 2.7 แสดงลักษณะการเรียนรู้โดยไม่อาศัยการแนะนำ

2.5 กฎการเรียนรู้ของเครือข่ายนิวรอน

ในที่นี้จะพิจารณาในกรณีของการเรียนรู้ที่อาศัยการแนะนำเท่านั้น เนื่องจากเป็นที่นิยมและนำมาใช้ในการออกแบบตัวควบคุมซึ่งจะกล่าวถึงในบทต่อไป สามารถแสดงการพิจารณากฎการเรียนรู้ได้ดังรูปที่ 2.8



***รูปที่ 2.8** แสดงการปรับค่าน้ำหนักโดยกฎการเรียนรู้ของเครือข่ายนิวรอน*

กฎการเรียนรู้ใช้ในการปรับค่าพารามิเตอร์ของเครือข่ายนิวรอนได้แก่ ค่าน้ำหนักและค่าไบแอส เพื่อให้เครือข่ายนิวรอนสร้างสัญญาณเอาต์พุตได้สอดคล้องกับชุดข้อมูลที่ได้เรียนรู้

<u>กฎการเรียนรู้โดยทั่วไป (General Learning Rule)</u>

กำหนดให้

- W ค่าน้ำหนักและค่าไบแอส
- X สัญญาณอินพุต
- r สัญญาณที่ใช้ในการเรียนรู้ (learning Signal)
- d สัญญาณที่ใช้ในการสอน (Teachering Signal)
- o สัญญาณเอาต์พุตจากปมนิวรอน
- α ค่าคงที่ของการเรียนรู้ (Learning Constant)
- k ลำดับเวลา (Time Step)

จากรูปที่ 2.8 สามารถเขียนสมการรูปทั่วไปของสัญญาณที่ใช้ในการเรียนรู้ได้ดังสมการที่ 2.7

$$r = r(W, X, d) \qquad (2.7)$$

ค่าน้ำหนักของเครือข่ายนิวรอนที่เปลี่ยนไป หลังจากได้รับสัญญาณที่ใช้ในการเรียนรู้ สามารถเขียนได้ดังสมการที่ 2.8

$$\Delta W(k) = \alpha . r\big(W(k), X(k), d(k)\big) X(k) \qquad (2.8)$$

ดังนั้นในทุกๆรอบของกระบวนการเรียนรู้ ค่าน้ำหนักของเครือข่ายนิวรอนจะถูกปรับเปลี่ยนให้เป็นค่าใหม่อยู่เสมอ สามารถแสดงได้ดังสมการที่ 2.9

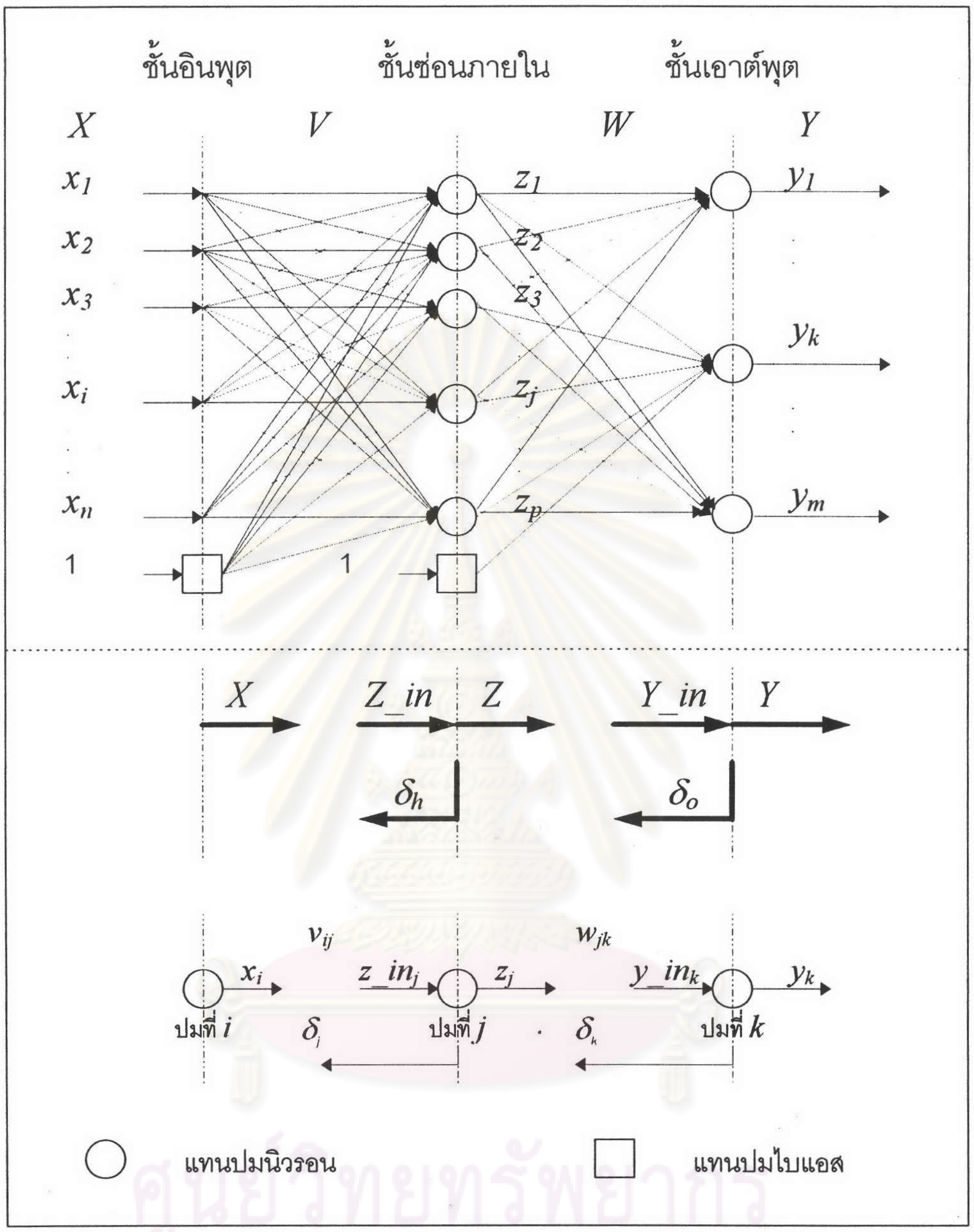
$$\Delta W(k+1) = W(k) + \Delta W(k) \qquad (2.9)$$

ชนิดของสัญญาณที่ใช้ในการเรียนรู้ อาศัยกฎที่มีลักษณะแตกต่างกันออกไป อาทิเช่น Hebbian Learning Rule, Perceptron Learning Rule, Delta Learning Rule ฯลฯ ในการเลือกใช้กฎใดขึ้นกับลักษณะของงานที่นำไปประยุกต์ใช้

ในที่นี้จะพิจารณาเครือข่ายนิวรอนชนิด Backpropagation ซึ่งเป็นกฎการเรียนรู้ปรับปรุงดัดแปลงจาก Delta Learning Rule ซึ่งนำไปใช้ในระบบควบคุมซึ่งจะกล่าวถึงต่อไป

2.6 <u>เครือข่ายนิวรอนชนิด Backpropagation</u>

เครือข่ายนิวรอนชนิด Backpropagation เป็นเครือข่ายนิวรอนที่ใช้วิธีการประมวลผลเป็นชั้นๆ โดยนำข้อมูลจากชุดข้อมูลที่ต้องการใช้ในการฝึกเครือข่าย (Training Set) ป้อนข้อมูลเข้าที่ชั้นอินพุต (Input Layer) ผ่านชั้นซ่อนภายใน (Hidden Layers) ถึงชั้นเอาต์พุต (Output Layer) สัญญาณออกจากชั้นเอาต์พุตถูกนำไปเปรียบเทียบกับสัญญาณตามที่ต้องการ (Desired Signal) หรือสัญญาณเป้าหมาย (Target Signal) ซึ่งได้จากชุดของข้อมูลที่นำมาใช้ฝึกเครือข่ายนิวรอน ค่าแตกต่างระหว่างสัญญาณทั้งสองถูกแผ่ย้อนกลับเข้าสู่เครือข่ายนิวรอนอีกครั้ง โดยผ่านทางชั้นเอาต์พุต (Output Layer) ผ่านชั้นซ่อนภายใน (Hidden Layers) จนถึงชั้นอินพุต (Input Layer) เพื่อปรับค่าพารามิเตอร์ของเครือข่ายนิวรอนในแต่ละชั้น อันได้แก่ ค่าน้ำหนัก ค่าไบแอส และค่าพารามิเตอร์ต่างๆของกฎที่ใช้ในการเรียนรู้ ทำให้เครือข่ายนิวรอนสามารถเรียนรู้ชุดข้อมูลที่แฝงมาในรูปของค่าพารามิเตอร์ของเครือข่าย และสามารถสร้างสัญญาณออกซึ่งสอดคล้องกับข้อมูลที่ใช้ในการฝึกเครือข่าย แสดงลักษณะโครงสร้างของเครือข่ายนิวรอนชนิด Backpropagation ในที่นี้พิจารณาในกรณี 3 ชั้น ดังรูปที่ 2.9



รูปที่ 2.9 *แสดงโครงสร้างของเครือข่ายนิวรอนชนิด Backpropagation*

<u>กำหนดให้</u>

X เวกเตอร์ของชุดข้อมูลอินพุต (Input Training Vector) ขนาด $n \times 1$

$$X = \begin{bmatrix} x_1 & \dots & x_i & \dots & x_n \end{bmatrix}^T$$

โดยที่ x_i แทนข้อมูลอินพุตที่ป้อนเข้าสู่ปมนิวรอนที่ i ในชั้นอินพุต

T เวกเตอร์ของชุดข้อมูลเป้าหมาย (Output Target Vector) ขนาด $m \times 1$

$$T = \begin{bmatrix} t_1 & \dots & t_k & \dots & t_m \end{bmatrix}^T$$

โดยที่ t_k แทนข้อมูลเป้าหมายของปมนิวรอนที่ k ในชั้นเอาต์พุต

Z เวกเตอร์ของสัญญาณออกจากปมนิวรอนในชั้นซ่อนภายใน ขนาด $p \times 1$

$$Z = \begin{bmatrix} z_1 & \dots & z_j & \dots & z_p \end{bmatrix}^T$$

โดยที่ z_j แทนสัญญาณออกจากปมนิวรอลที่ j ในชั้นซ่อนภายใน

Y เวกเตอร์ของสัญญาณออกจากปมนิวรอนในชั้นเอาต์พุต ขนาด $m \times 1$

$$Y = \begin{bmatrix} y_1 & \dots & y_k & \dots & y_m \end{bmatrix}^T$$

โดยที่ y_k แทนสัญญาณออกจากปมนิวรอลที่ k ในชั้นเอาต์พุต

V เมตริกซ์ของค่าน้ำหนักระหว่างชั้นอินพุตกับชั้นซ่อนภายใน ขนาด $n \times p$

$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1p} \\ v_{21} & v_{22} & \cdots & v_{2p} \\ \cdots & \cdots & v_{ij} & \cdots \\ v_{n1} & v_{n2} & \cdots & v_{np} \end{bmatrix}$$

โดยที่ v_{ij} แทนค่าน้ำหนักที่เชื่อมระหว่างปมนิวรอนที่ i ในชั้นอินพุตกับปมนิวรอนที่ j ในชั้นซ่อนภายใน

V_{Bias} เวกเตอร์ของค่าไบแอสของชั้นซ่อนภายใน ขนาด $p \times 1$

$$V_{Bias} = \begin{bmatrix} v_1 & \cdots & v_j & \cdots & v_p \end{bmatrix}^T$$

โดยที่ v_j แทนค่าไบแอสของปมนิวรอนที่ j ในชั้นซ่อนภายใน

W เมตริกซ์ของค่าน้ำหนักระหว่างชั้นซ่อนภายในและชั้นเอาต์พุต ขนาด $p \times m$

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \cdots & \cdots & w_{jk} & \cdots \\ w_{p1} & w_{p2} & \cdots & w_{pm} \end{bmatrix}$$

โดยที่ w_{jk} แทนค่าน้ำหนักที่เชื่อมระหว่างปมนิวรอนที่ j ในชั้นซ่อนภายในกับปมนิวรอนที่ k ในชั้นเอาต์พุต

W_{bias} เวกเตอร์ของค่าไบแอสของชั้นเอาต์พุต ขนาด $m \times 1$

$$W_{Bias} = \left[w_1 \quad \dots \quad w_k \quad \dots \quad w_m\right]^T$$

โดยที่ w_k แทนค่าไบแอสของปมนิวรอนที่ k ในชั้นเอาต์พุต

δ_o เวกเตอร์สัญญาณคลาดเคลื่อนของปมนิวรอนในชั้นเอาต์พุต ขนาด $m \times 1$

$$\delta_o = \left[\delta_1 \quad \dots \quad \delta_k \quad \dots \quad \delta_m\right]^T$$

โดยที่ δ_k แทนสัญญาณคลาดเคลื่อนของปมนิวรอนที่ k ในชั้นเอาต์พุต

δ_h เวกเตอร์สัญญาณคลาดเคลื่อนของปมนิวรอนในชั้นซ่อนภายใน ขนาด $p \times 1$

$$\delta_h = \left[\delta_1 \quad \dots \quad \delta_j \quad \dots \quad \delta_p\right]^T$$

โดยที่ δ_j แทนสัญญาณคลาดเคลื่อนของปมนิวรอนที่ j ในชั้นซ่อนภายใน

Z_in เวกเตอร์ของค่าสัญญาณที่เข้าสู่ชั้นซ่อนภายใน ขนาด $p \times 1$

$$Z\_in = \left[z\_in_1 \quad \dots \quad z\_in_j \quad \dots \quad z\_in_p\right]^T$$

โดยที่ z_in_j แทนค่าสัญญาณเข้าปมนิวรอนที่ j ในชั้นซ่อนภายใน

Y_in เวกเตอร์ของค่าสัญญาณที่เข้าสู่ชั้นเอาต์พุต ขนาด $m \times 1$

$$Y\_in = \left[y\_in_1 \quad \dots \quad y\_in_k \quad \dots \quad y\_in_m\right]^T$$

โดยที่ y_in_k แทนค่าสัญญาณเข้าปมนิวรอนที่ k ในชั้นเอาต์พุต

α ค่าอัตราการเรียนรู้ (Learning Rate)

ลำดับขั้นการประมวลผลของเครือข่ายนิวรอน

ขั้นที่ 1 ชุดข้อมูลที่ใช้ในการฝึกเครือข่ายนิวรอน (Training Set)

ในการเรียนรู้ของเครือข่ายนิวรอนโดยอาศัยการแนะนำ จำเป็นต้องใช้ชุดข้อมูลอินพุต และข้อมูลเป้าหมายในการฝึกเครือข่ายนิวรอน เพื่อให้เครือข่ายดังกล่าวสร้างสัญญาณออกได้สอดคล้องกับสัญญาณเป้าหมายเมื่อเครือข่ายนิวรอนได้รับข้อมูลสัญญาณเข้าลักษณะเดียวกัน ชุดของข้อมูลที่ใช้ในการฝึกเครือข่ายสามารถเขียนแทนได้ดังสมการที่ 2.10

$$P = \{(X(1), T(1)), (X(2), T(2)), \ldots, (X(N), T(N))\} \quad \text{.................. (2.10)}$$

โดยที่ P ชุดของข้อมูลที่ใช้ในการฝึกเครือข่าย (Training Set)

N จำนวนชุดข้อมูลที่ใช้ในการฝึกเครือข่าย (Training Pair)

ขั้นที่ 2 การกำหนดค่าเริ่มต้นสำหรับพารามิเตอร์ของเครือข่ายนิวรอน

การฝึกเครือข่ายนิวรอนจำเป็นต้องกำหนดค่าเริ่มต้นให้กับค่าน้ำหนัก (Weight) และค่าไบแอส (Bias) รวมทั้งค่าอัตราการเรียนรู้ (Learning rate) เพื่อใช้เป็นจุดเริ่มต้นในกระบวนการประมวลผลของเครือข่ายซึ่งจะกล่าวต่อไป

ขั้นที่ 3 กระบวนการประมวลผลของเครือข่าย

กระบวนการประมวลผลของเครือข่ายนิวรอนแบ่งออกเป็น 2 ตอน

ตอนที่ 1 การคำนวนไปข้างหน้า (Feedforward Calculation)

การคำนวนเริ่มจากชั้นอินพุต ผ่านชั้นซ่อนภายใน จนถึงชั้นเอาต์พุต แสดงได้ดังสมการที่ 2.11-2.14

พิจารณาชั้นอินพุต (Input Layer)

ปมนิวรอนในชั้นอินพุตจะรับข้อมูลอินพุต (X) จากชุดข้อมูลที่ใช้ในการฝึกเครือข่าย (Trainning Set) และส่งต่อไปยังชั้นภายในเริ่มทำการคำนวนต่อไป

พิจารณาชั้นซ่อนภายใน (Hidden Layer)

สัญญาณที่เข้าสู่ชั้นซ่อนภายในสามารถแสดงได้ดังสมการที่ 2.11

$$Z\_in = V^T X + V_{Bias} \quad \text{..........} (2.11)$$

สัญญาณที่ออกจากชั้นซ่อนภายในสามารถแสดงได้ดังสมการที่ 2.12

$$Z = f(Z\_in) \quad \text{..........} (2.12)$$

พิจารณาชั้นเอาต์พุต (Output Layer)

สัญญาณที่เข้าสู่ชั้นเอาต์พุตสามารถแสดงได้ดังสมการที่ 2.13

$$Y\_in = W^T Z + W_{Bias} \quad \text{..........} (2.13)$$

สัญญาณที่ออกจากชั้นเอาต์พุตสามารถแสดงได้ดังสมการที่ 2.14

$$Y = f(Y\_in) \quad \text{..........} (2.14)$$

ตอนที่ 2 การแผ่ค่าความคลาดเคลื่อนย้อนกลับ (Backpropagation of Error)

การปรับค่าน้ำหนักของเครือข่ายนิวรอนเพื่อลดค่าความคลาดเคลื่อนหรือความแตกต่างระหว่างข้อมูลเป้าหมาย $T(k)$ กับสัญญาณออกจากเครือข่ายนิวรอนในชั้นเอาต์พุต $Y(k)$ ใช้วิธีการออพติไมซ์แบบเกรเดียนท์ โดยกำหนดฟังก์ชันความคลาดเคลื่อน (Error Function : $E(k)$) ดังแสดงในสมการที่ 2.15 ในที่นี้พิจารณาในกรณีปมนิวรอนในชั้นเอาต์พุต

มีจำนวน 1 ปม เนื่องจากระบบควบคุมที่นำไปใช้เป็นระบบอินพุตเอาต์พุตเดียว (Single Input and Single Output : SISO Syetem)

$$E(k) = \frac{1}{2}\left[T(k) - Y(k)\right]^2 \quad \text{.......................................(2.15)}$$

เมื่อ $E(k)$ แทนฟังก์ชันความคลาดเคลื่อนของชุดข้อมูลที่ k

คำนวนการลดค่าความคลาดเคลื่อนเทียบกับค่าน้ำหนัก และค่าไบแอสของเครือข่ายนิวรอนที่เชื่อมอยู่ระหว่างชั้นซ่อนภายในกับชั้นเอาต์พุต แสดงในสมการที่ 2.16

$$\frac{\partial E}{\partial W} = \frac{\partial\left[0.5\left(T(k) - Y(k)\right)^2\right]}{\partial W}$$

$$= \frac{\partial\left[0.5\left(T(k) - f(Y\_in)\right)^2\right]}{\partial W}$$

$$= -\left[T(k) - Y(k)\right]\frac{\partial f\left(Y\_in\right)}{\partial W}$$

$$= -\left[T(k) - Y(k)\right]f'(Y\_in)\frac{\partial\left(Y\_in\right)}{\partial W}$$

$$\therefore \quad \frac{\partial E}{\partial W} = -\left[T(k) - Y(k)\right]f'(y\_in)Z \quad \text{...(2.16)}$$

พิจารณาชั้นเอาต์พุต (Output Layer)

จากสมการที่ 2.16 กำหนด δ_o เป็นความสัมพันธ์แสดงดังสมการที่ 2.17

$$\delta_o = \left[T(k) - Y(k)\right]f'(Y\_in) \quad \text{.......................................(2.17)}$$

การเปลี่ยนแปลงค่าน้ำหนักของแขนซึ่งเชื่อมระหว่างชั้นซ่อนภายในกับชั้นเอาต์พุต (ΔW) แสดงดังสมการที่ 2.18 - 2.19

$$\Delta W = -\alpha \frac{\partial E}{\partial W} \quad \text{.......(2.18)}$$

หรือ

$$\Delta W = \alpha Z \delta_o^T \quad \text{.......(2.19)}$$

ในทำนองเดียวกัน การเปลี่ยนแปลงค่าไบแอสของปมนิวรอนในชั้นเอาต์พุต (ΔW_{Bias}) แสดงดังสมการที่ (2.20)

$$\Delta W_{Bias} = \alpha \delta_o \quad \text{.......(2.20)}$$

พิจารณาชั้นซ่อนภายใน (Hidden Layer)

การลดค่าความคลาดเคลื่อนเทียบกับค่าน้ำหนักและค่าไบแอสที่เชื่อมระหว่างชั้นอินพุตกับชั้นซ่อนภายในพิจารณาทำนองเดียวกับชั้นเอาต์พุต ดังแสดงในสมการที่ 2.21

$$\begin{aligned}
\frac{\partial E}{\partial V} &= \frac{\partial\left[0.5(T(k)-Y(k))^2\right]}{\partial V} \\
&= -\left[T(k)-Y(k)\right]\frac{\partial Y(k)}{\partial V} \\
&= -\left[T(k)-Y(k)\right]f'(Y_in)\frac{\partial Y_in}{\partial V} \\
&= -\delta_o \frac{\partial Y_in}{\partial V} \\
&= -\delta_o W \frac{\partial Z}{\partial V}
\end{aligned}$$

$$\therefore \quad \frac{\partial E}{\partial V} = -\delta_o W f'(Z\_in) X \quad \text{.......(2.21)}$$

เมื่อกำหนดให้ δ_h เป็นความสัมพันธ์แสดงดังสมการที่ 2.22

$$\delta_h = (W\delta_o).f'(Z\_in) \quad \text{.......(2.22)}$$

การเปลี่ยนแปลงค่าน้ำหนักที่เชื่อมระหว่างชั้นอินพุตกับชั้นซ่อนภายใน (ΔV) แสดงดังสมการที่ 2.23 - 2.24

$$\Delta V = -\alpha \frac{\partial E}{\partial V} \quad (2.23)$$

หรือ

$$\Delta V = \alpha X \delta_h^T \quad (2.24)$$

ในทำนองเดียวกัน การเปลี่ยนแปลงค่าไบแอสของปมนิวรอนในชั้นซ่อนภายใน (ΔV_{Bias}) แสดงดังสมการที่ (2.25)

$$\Delta V_{Bias} = \alpha \delta_h \quad (2.25)$$

<u>ขั้นที่ 4</u> การปรับค่าน้ำหนัก (Weight) และค่าไบแอส (Bias)

หลังจากการประมวลผลในขั้นที่ 3 เครือข่ายนิวรอนจะปรับฐานข้อมูลความรู้ภายในใหม่ ซึ่งอยู่ในรูปของการปรับค่าน้ำหนัก และค่าน้ำหนักไบแอสในแต่ละชั้นให้ทันสมัยและสอดคล้องกับชุดข้อมูลความรู้ใหม่ที่ได้เรียนรู้ เป็นผลให้ฟังก์ชันความคลาดเคลื่อนดังสมการที่ 2.15 มีค่าลดลง

พิจารณาการปรับค่าน้ำหนัก และค่าไบแอสของแขนที่เชื่อมระหว่างชั้นซ่อนภายในกับชั้นเอาต์พุต

แสดงดังสมการที่ 2.26 และ 2.27 ตามลำดับ

$$\Delta V(k+1) = V(k) + \Delta V(k) \quad (2.26)$$

$$\Delta V_{Bias}(k+1) = V_{Bias}(k) + \Delta V_{Bias}(k) \quad (2.27)$$

พิจารณาการปรับค่าน้ำหนัก และค่าไบแอสของแขนที่เชื่อมระหว่างชั้นซ่อนภายในกับชั้นเอาต์พุต

แสดงดังสมการที่ 2.28 และ 2.29 ตามลำดับ

$$\Delta W(k+1) = W(k) + \Delta W(k) \quad \text{.......(2.28)}$$

$$\Delta W_{Bias}(k+1) = W_{Bias}(k) + \Delta W_{Bias}(k) \quad \text{.......(2.29)}$$

<u>ขั้นที่ 5</u> ตรวจสอบจำนวนชุดข้อมูลที่ใช้ฝึกเครือข่าย

ข้อมูลที่ใช้ในการฝึกเครือข่าย (Training Set : P) จะถูกป้อนเข้ามาครั้งละ 1 คู่ลำดับ (Training Pair : $(X(k), T(k))$) ซึ่งประกอบไปด้วย ข้อมูลอินพุต (Input Data : $X(k)$) และข้อมูลเป้าหมาย (Target Data : $T(k)$) แต่เนื่องจากชุดข้อมูลที่ใช้ในการฝึกเครือข่ายมีจำนวนทั้งสิ้น N ชุด ดังนั้นจำเป็นต้องตรวจสอบจำนวนข้อมูลที่ใช้ในการฝึกเครือข่ายได้รับมาครบหรือไม่ ถ้าไม่ครบตามจำนวนจำเป็นต้องป้อนข้อมูลชุดถัดมา แล้วจึงเริ่มทำการประมวลผลตามขั้นตอนที่ 3 อีกครั้ง แต่ถ้าจำนวนข้อมูลที่ใช้ฝึกรับมาครบตามจำนวนจะผ่านไปยังขั้นตอนที่ 6 ต่อไป

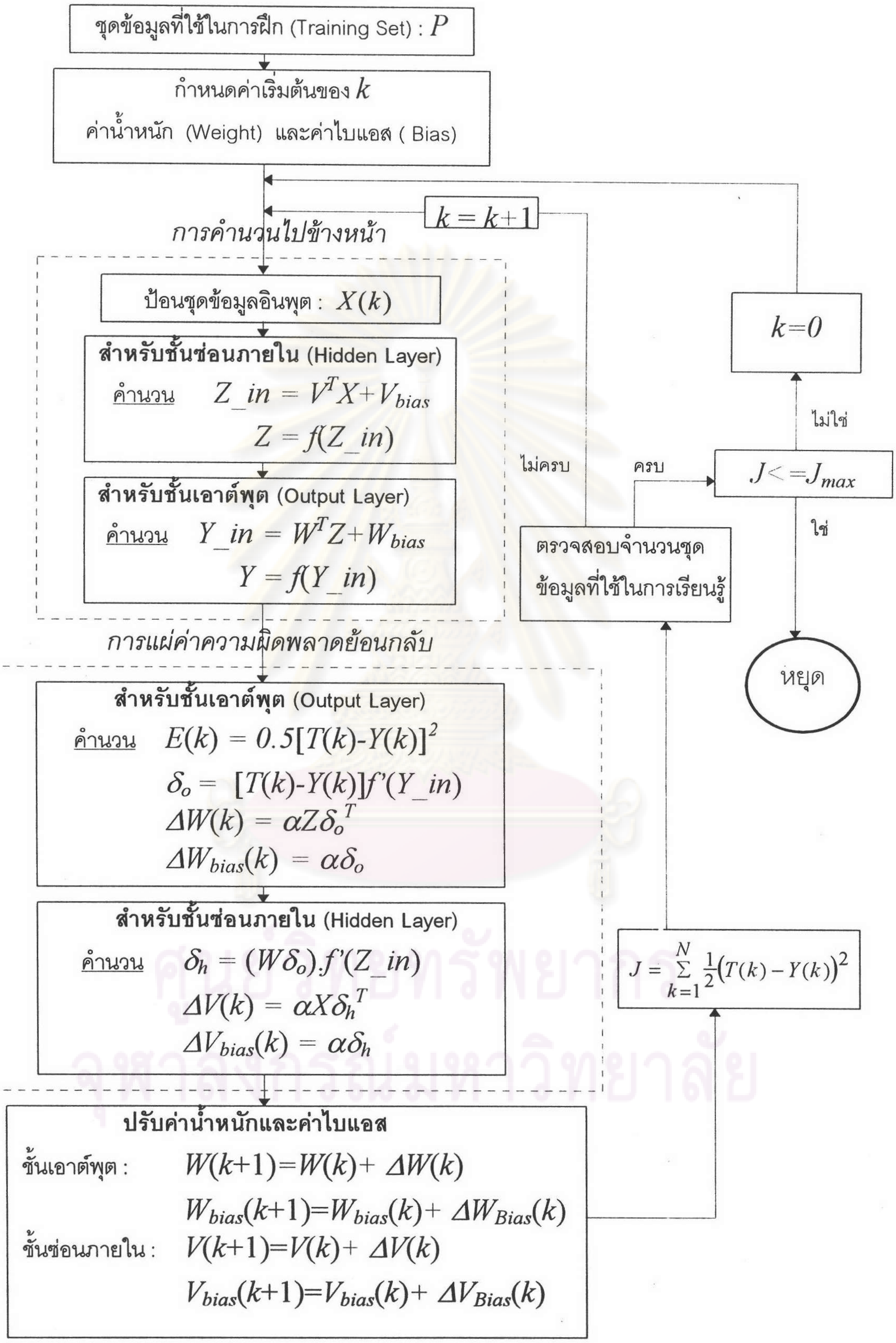
<u>ขั้นที่ 6</u> ตรวจสอบดรรชนีสมรรถนะ (Performance Index)

วัตถุประสงค์หลักของการฝึกเครือข่ายนิวรอนคือ การลดฟังก์ชันความคลาดเคลื่อนระหว่างข้อมูลที่ออกจากเครือข่ายนิวรอนและข้อมูลเป้าหมาย ดรรชนีสมรรถนะของการฝึกเครือข่ายนิวรอนแสดงดังสมการที่ 2.30

$$J = \sum_{k=1}^{N} \frac{1}{2}\left(T(k) - Y(k)\right)^2 \quad \text{.......(2.30)}$$

โดยที่ J ดรรชนีสมรรถนะของการฝึกเครือข่ายนิวรอน

ถ้าดรรชนีสมรรถนะ (J) มีค่าต่ำกว่าที่กำหนด ($J_{\max}$) แสดงว่าเครือข่ายนิวรอนได้รับการฝึกให้สามารถเรียนรู้และสามารถสร้างผลตอบได้สอดคล้องกับข้อมูลที่ป้อนให้เรียนรู้ ดังนั้นจึงหยุดการฝึกเครือข่ายและพร้อมที่จะนำเครือข่ายนิวรอนดังกล่าวไปประยุกต์ใช้งานต่อไป ในทางตรงกันข้ามถ้าดรรชนีสมรรถนะ (J) มีค่ามากกว่าที่กำหนด ($J_{\max}$) จะเริ่มทำการฝึกตามขั้นที่ 3 อีกครั้ง จากการประมวลผลทั้ง 6 ขั้นตอน สามารถสรุปได้ดังรูปที่ 2.10



รูปที่ 2.10 *แสดงลำดับขั้นการประมวลผลของเครือข่ายนิวรอน*

2.7 การปรับปรุงอัตราการเรียนรู้โดยวิธี Delta-Bar-Delta

โดยทั่วไปวิธีการปรับค่าน้ำหนักและค่าไบแอสของเครือข่ายนิวรอนดังกล่าวในหัวข้อที่ 2.6 ใช้วิธี Steepest Descent ซึ่งค่าอัตราการเรียนรู้ α จะถูกกำหนดเป็นค่าตายตัว ดังนั้นจึงเป็นการยากที่จะเลือกค่าอัตราการเรียนรู้ที่เหมาะสมสำหรับใช้ในการฝึกเครือข่ายนิวรอน ดังนั้นจึงมีการพัฒนาอัลกอลิทึมเพื่อให้การเรียนรู้ของเครือข่ายนิวรอลเป็นไปได้อย่างรวดเร็ว เรียกว่า วิธี *Delta-Bar-Delta* โดยการนำเสนอของ Jacob [11] หลักการทั่วไปคือ ต้องการให้ค่าน้ำหนักและค่าไบแอสที่เชื่อมระหว่างปมนิวรอนในแต่ละชั้นมีอัตราการเรียนรู้เป็นของตัวเอง และถ้าการเปลี่ยนแปลงค่าน้ำหนักและค่าไบแอสเป็นไปในทิศทางที่ทำให้ฟังก์ชันความคลาดเคลื่อนลดลง สังเกตุได้จากการเปลี่ยนแปลงเครื่องหมายอนุพันธ์ย่อยของฟังก์ชันความคลาดเคลื่อนเทียบกับค่าน้ำหนักมีเครื่องหมายเดียวกัน ดังนั้นค่าอัตราการเรียนรู้จะมีค่าเพิ่มขึ้นในลักษณะเชิงเส้นเพื่อเป็นการป้องกันค่าอัตราการเรียนรู้ที่อาจจะเพิ่มมากขึ้นอย่างรวดเร็วเกินไป ในทางกลับกันถ้าการเปลี่ยนแปลงค่าน้ำหนักและค่าไบแอสเปลี่ยนไปในทางตรงกันข้าม แสดงว่าค่าน้ำหนักและค่าไบแอสเริ่มมีค่าใกล้เคียงกับค่าน้ำหนักและค่าไบแอสที่เหมาะสมที่ทำให้ฟังก์ชันความคลาดเคลื่อนลดลงต่ำสุด อัตราการเรียนรู้จะมีค่าลดลง แสดงอัลกอลิทึมการปรับอัตราการเรียนรู้โดยวิธี Delta-Bar-Delta ได้ดังนี้

<u>กำหนดให้</u>

$\alpha_W(k)$ — เมตริกซ์อัตราการเรียนรู้ของค่าน้ำหนัก (Weight) ที่เชื่อมระหว่างชั้นซ่อนภายในกับชั้นเอาต์พุต ที่เวลา k ขนาด $p \times m$

$\alpha_{W_{Bias}}(k)$ — เมตริกซ์อัตราการเรียนรู้ของค่าไบแอส (Bias) ที่เชื่อมระหว่างชั้นซ่อนภายในกับชั้นเอาต์พุต ที่เวลา k ขนาด $m \times 1$

$\alpha_V(k)$ — เมตริกซ์อัตราการเรียนรู้ของค่าน้ำหนัก (Weight) ที่เชื่อมระหว่างชั้นอินพุตกับชั้นซ่อนภายใน ที่เวลา k ขนาด $n \times p$

$\alpha_{V_{Bias}}(k)$ — เมตริกซ์อัตราการเรียนรู้ของค่าน้ำหนักไบแอส (Bias) ที่เชื่อมระหว่างชั้นอินพุตกับชั้นซ่อนภายใน ที่เวลา k ขนาด $p \times 1$

δ_o — เวกเตอร์สัญญาณคลาดเคลื่อนของปมนิวรอนในชั้นเอาต์พุต

δ_h — เวกเตอร์สัญญาณคลาดเคลื่อนของปมนิวรอนในชั้นซ่อนภายใน

$\gamma, \beta, \mathrm{K}$ — ค่าคงที่ซึ่งใช้ในการปรับอัตราการเรียนรู้ โดยวิธี Delta-Bar-delta โดยที่ $0< \beta <1$

<u>พิจารณาชั้นเอาต์พุต</u>

การเปลี่ยนแปลงค่าน้ำหนัก (Weight : W)

แสดงได้ดังสมการที่ 2.31 หรือ 2.32

$$W(k+1) = W(k) - \alpha_W(k+1)\frac{\partial E}{\partial W} \quad \text{.........(2.31)}$$

หรือ

$$W(k+1) = W(k) + \alpha_W(k+1) Z\delta_o^T \quad \text{.........(2.32)}$$

จากสมการข้างต้น กำหนดให้

$$\underset{w}{\Delta}(k) = \frac{\partial E}{\partial W} = -Z\delta_o^T \quad \text{.........(2.33)}$$

$$\underset{w}{\overline{\Delta}}(k) = (1-\beta)\underset{w}{\Delta}(k) + \beta\underset{w}{\overline{\Delta}}(k-1) \quad \text{.........(2.34)}$$

อัตราการเรียนรู้ของค่าน้ำหนักซึ่งเชื่อมระหว่างชั้นซ่อนภายในกับชั้นเอาต์พุตที่เปลี่ยนแปลงไป ($\alpha_W(k+1)$) แสดงดังสมการที่ 2.35

$$\alpha_W(k+1) = \begin{cases} \alpha_W(k) + \mathrm{K} & ; \underset{w}{\overline{\Delta}}(k-1)\underset{w}{\Delta}(k) > 0 \\ (1-\gamma)\alpha_W(k) & ; \underset{w}{\overline{\Delta}}(k-1)\underset{w}{\Delta}(k) < 0 \\ \alpha_W(k) & ; \text{Otherwise} \end{cases} \quad \text{.........(2.35)}$$

การเปลี่ยนแปลงค่าไบแอส (Bias : W_{Bias})

พิจารณาทำนองเดียวกับการเปลี่ยนแปลงค่าน้ำหนักข้างต้น แสดงการเปลี่ยนแปลงค่าไบแอสได้ดังสมการที่ 2.36-2.37

$$W_{Bias}(k+1) = W_{Bias}(k) - \alpha_{W_{Bias}}(k+1)\frac{\partial E}{\partial W_{Bias}} \quad \text{................(2.36)}$$

หรือ

$$W_{Bias}(k+1) = W_{Bias}(k) - \alpha_{W_{Bias}}(k+1)\delta_o \quad \text{..........................(2.37)}$$

จากสมการข้างต้น กำหนดให้

$$\underset{w_{Bias}}{\Delta}(k) = \frac{\partial E}{\partial W_{Bias}} = -\delta_o \quad \text{..(2.38)}$$

$$\underset{w_{Bias}}{\overline{\Delta}}(k) = (1-\beta)\underset{w_{Bias}}{\Delta}(k) + \beta\,\overline{\underset{w_{Bias}}{\Delta}}(k-1) \quad \text{.............................(2.39)}$$

คำนวนอัตราการเรียนรู้ของค่าไบแอสที่เปลี่ยนแปลงไป ($\alpha_{W_{Bias}}(k+1)$) ได้เช่นเดียวกับสมการที่ 2.35 โดยเปลี่ยนจากค่าน้ำหนักเป็นค่าไบแอส

<u>พิจารณาชั้นซ่อนภายใน</u>

การเปลี่ยนแปลงค่าน้ำหนัก (Weight : V)

แสดงดังสมการที่ 2.40-2.41

$$V(k+1) = V(k) - \alpha_V(k+1)\frac{\partial E}{\partial V} \quad \text{...(2.40)}$$

หรือ

$$V(k+1) = V(k) + \alpha_V(k+1)X\delta_h^T \quad \text{...(2.41)}$$

จากสมการข้างต้น กำหนดให้

$$\underset{v}{\Delta}(k) = \frac{\partial E}{\partial V} = -X\delta_h^T \quad \text{...(2.42)}$$

$$\underset{v}{\overline{\Delta}}(k) = (1-\beta)\underset{v}{\Delta}(k) + \beta\underset{v}{\overline{\Delta}}(k-1) \quad \text{..(2.43)}$$

อัตราการเรียนรู้ของค่าน้ำหนักซึ่งเชื่อมระหว่างชั้นอินพุตและชั้นซ่อนภายในที่เปลี่ยนแปลงไป ($\alpha_V(k+1)$) แสดงได้เช่นเดียวกับสมการที่ 2.35

การเปลี่ยนแปลงค่าน้ำหนักไบแอส (Bias : V_{bias})

พิจารณาทำนองเดียวกับการเปลี่ยนแปลงค่าน้ำหนักข้างต้น แสดงดังสมการที่ 2.44-2.45

$$V_{Bias}(k+1) = V_{Bias}(k) - \alpha_{V_{Bias}}(k+1)\frac{\partial E}{\partial V_{Bias}} \quad \text{..................(2.44)}$$

หรือ

$$V_{Bias}(k+1) = V_{Bias}(k) - \alpha_{V_{Bias}}(k+1)\delta_h \quad \text{...........................(2.45)}$$

จากสมการข้างต้น กำหนดให้

$$\underset{v_{Bias}}{\Delta}(k) = \frac{\partial E}{\partial V_{Bias}} = -\delta_h \quad \text{...(2.46)}$$

$$\underset{v_{Bias}}{\overline{\Delta}}(k) = \left(1-\beta\right)\underset{v_{Bias}}{\Delta}(k) + \beta\,\overline{\underset{v_{Bias}}{\Delta}}(k-1) \quad \text{...............................(2.47)}$$

อัตราการเรียนรู้ของค่าไบแอสซึ่งเชื่อมระหว่างชั้นอินพุตและชั้นซ่อนภายในที่เปลี่ยนแปลงไป ($\alpha_{V_{Bias}}(k+1)$) แสดงได้เช่นเดียวกับสมการที่ 2.35 โดยเปลี่ยนจากค่าน้ำหนักเป็นค่าไบแอส และคำนวนในทำนองเดียวกัน

จากการศึกษาความรู้พื้นฐานและลักษณะโดยทั่วไปของเครือข่ายนิวรอนดังกล่าวถึงในบทนี้ ได้นำเครือข่ายนิวรอนชนิด Backpropagation ซึ่งใช้วิธีการปรับค่าน้ำหนักและค่าไบแอสตามวิธี Delta-Bar-Delta เนื่องจากเป็นเครือข่ายนิวรอนที่อาศัยการเรียนรู้แบบโดยอาศัยการแนะนำ (Supervised Learning) ดังนั้นจึงสามารถฝึกเครือข่ายให้มีความสามารถในการสร้างความสัมพันธ์จากชุดข้อมูลที่ใช้ในการฝึกเครือข่าย ซึ่งนำไปประยุกต์ใช้ในระบบควบคุมแบบปรับตัวเองตามแบบจำลองอ้างอิงดังจะกล่าวถึงในบทที่ 3 ต่อไป